2020-12

CHOICE CONFIDENCE AND PERSUASION RESISTANCE THROUGH MOUSE ACTION OBSERVATION

Cheer, Harlan T.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CHOICE CONFIDENCE AND PERSUASION RESISTANCE THROUGH MOUSE ACTION OBSERVATION

by

Harlan T. Cheer

December 2020

Thesis Advisor: Vinnie Monaco
Second Reader: Marko Orescanin

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE CHOICE CONFIDENCE AND PERSUASION RESISTANCE THROUGH MOUSE ACTION OBSERVATION	**5. FUNDING NUMBERS**
6. AUTHOR(S) Harlan T. Cheer	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

With the growing interconnected nature of social discourse, manipulative actors take advantage of those who are more open to persuasion. One way of determining those who are susceptible to manipulation are those who do not hold concrete opinions on a subject and may take the word of those who wish to direct their position on social or political issues. One way this could be done is if a person's confidence on a topic could be evaluated and then subjugated to directed propaganda with the goal to change or solidify their choice. By conducting an online survey with mouse tracking analytics, this thesis aims to determine how online behavior could be turned against a user to direct their activity online. With such data, privacy advocates and government regulators will be better equipped to evaluate internet data-gathering methods of people online in order to better protect people from potential threats of online groups that are looking to manage the behavior of others.

14. SUBJECT TERMS human computer interaction, deep learning, confidence detection	**15. NUMBER OF PAGES** 69
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

CHOICE CONFIDENCE AND PERSUASION RESISTANCE THROUGH MOUSE ACTION OBSERVATION

Harlan T. Cheer
Civilian, CyberCorps: Scholarship for Service
BS, California State University Monterey Bay, 2018

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2020

Approved by: Vinnie Monaco
Advisor

Marko Orescanin
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the growing interconnected nature of social discourse, manipulative actors take advantage of those who are more open to persuasion. One way of determining those who are susceptible to manipulation are those who do not hold concrete opinions on a subject and may take the word of those who wish to direct their position on social or political issues. One way this could be done is if a person's confidence on a topic could be evaluated and then subjugated to directed propaganda with the goal to change or solidify their choice. By conducting an online survey with mouse tracking analytics, this thesis aims to determine how online behavior could be turned against a user to direct their activity online. With such data, privacy advocates and government regulators will be better equipped to evaluate internet data-gathering methods of people online in order to better protect people from potential threats of online groups that are looking to manage the behavior of others.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

ADAM	Adaptive Moment Estimation
AUC	Area Under the Curve
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DPI	Dots Per Inch
GPU	Graphics Processing Unit
HCI	Human Computer Interaction
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MD	Max Deviation
RNN	Recurrent Neural Network

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

This material is based upon activities supported by the National Science Foundation under Agreement No 1565443. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The rapid expansion and simplification of communication have enabled rampant interconnection between people. Thoughts and opinions can be shared instantly, viewable by practically every person on Earth in a moment. On popular social media platforms, netizens worldwide, unrestricted by borders, share their thoughts and opinions on every topic imaginable. Social and political cliques form, where like-minded users bolster each other and reinforce their position while denigrating their perceived opposition. What happens when discourse becomes obfuscated and manipulated, as seen in the 2016 United States presidential election? Online opinion can be directed and exploited, which results in the spread of disinformation and discreditation of facts [1].

1.1 Problem Statement

New concerns from online actors working to subvert public opinion to sow discourse and distrust have emerged on social media platforms. These methods look to persuade unformed voters to join the manufactured online consensus, with the use of social media "trolls," bots, and political advertising [2], to convince a person to change or solidify their position, possibly working against their old position or reinforcing the one contrary to their opinion.

It is impossible to know the specific goals of online disinformation actors, but many believe that merely muddying the waters of social media is enough [1]. Niche and anti-intellectual topics have been given a megaphone, convincing the uninformed to promote these topics further. Ideas such as the belief in a flat-Earth, vaccine skepticism, and now seen with the COVID-19 pandemic, anti-PPE, and virus denial stances have been given a foundation in perceived social acceptance. While it cannot be said that these ideas would not naturally come about on their own, evidence suggests these ideas were given heightened perceptions, and therefore perceived legitimacy, by often called "Russian trolls" and other similar groups.

1.2 Research Objectives

This thesis examines a related discussion on the confidence of choice to see if their confidence can be measured and accurately predicted. There is a connection between lies and computer interaction, being able to ascertain if a human is telling the truth or not through their performance while using a computer [3]. This work aims to specifically examine if it is possible to determine if a person's response was given because they were sure of an answer or if guesswork or random choice is involved. The difference between the similar topic of lie detection is not to see if a person is honest, but if they can confidently respond to a question, accurately or not. Lie detection relates to the cognitive mechanisms involved in suppressing an honest reply and then constructing a false response. Measuring confidence differs in that the cognitive process involves not suppressing an answer but evaluating personal values and then choosing how to respond. A confident response involves quicker evaluation, while an unsure response takes more time to make a decision.

There are further conversations to be explored to understand how people online are subject to the information and opinions they are exposed to. Privacy on the web is a complex and tumultuous topic. Exploring the impact of user observation and data collection, and what is possible with such data, is an essential addition to this area. This work can open the way for further examination of persuasion measurement and detect a change of opinion effectively. This has impacts on advertising and political messaging, in addition to raising the issue of online privacy and how much of a person's digital activity should be examined and used to construct highly detailed profiles of a person's opinions and beliefs. There are other potential applications for cybersecurity. These metrics could also be used in digital observation and espionage, to help identify individuals who may be vulnerable to social engineering attacks.

1.3 Organization

This work's structure is first an overview of human-computer interaction, discussing lie detection, and observing the study of metacognitive behavior and how it relates to lying and choice confidence. Next is a discussion of the experimentation methodology, beginning with an introduction to MouseTracker and the data collected using several previous studies into choice confidence and temporal discounting. The development of a machine learning model analyzes the mouse trajectory data of subjects to determine quantitatively measure

the confidence in their responses. Then an analysis of the data and the model to discuss how well it can determine a response's confidence. The closing will discuss future work in this area and outline questions this work raises and how this approach might solve them.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Previous Work

The field of human-computer interaction (HCI) is an expanding field that covers many potential insights into human behavior. With the wide range of points of interaction with computers, much can be understood that is not readily apparent. We start with a survey of works on human choice confidence as an aspect of the study of metacognition in psychology, then examining related works in the areas of cognition measurement, biometric authentication, and lie detection using HCI metrics.

2.1 Cognition, Mood, and Device Dynamics

This section contains a survey into HCI research, including Fitt's Law, which describes how long it takes a user to perform a task with a mouse, and Hick's Law, which characterizes how long it takes to decide concerning how many choices are present to the user [4]. These laws provide an insight into human behavior when using mouse devices and inform how actions using a computer mouse may play out.

Years of research in HCI has found the importance and impact of giving computers the ability to collect, analyze, and understand user mental state. Early work in understanding how to interpret and leverage human action on machines [5] shows that the way that humans interact with computers exposes their internal state, allowing for the interpretation of human mood while using a computer. They were able to develop methods that used the keyboard and mouse to understand the human emotional state. Further research [6] reinforces these findings using mouse motion dynamics and selecting features using random forest techniques.

The prolific growth of HCI through Internet activity allows for even more opportunities to analyze human behavior. Web-based approaches to evaluating human mood and emotion have been achieved by recording mouse movements on a webpage [7]. From mouse dynamics and related HCI biometrics, researchers could "analyze a person's eleven states of being (stress, work productivity, mood, interest in work) and seven emotions (self-control, happiness, anger, fear, sadness, surprise, and anxiety) during a realistic timeframe." From

the user's determined "physiological, psychological, and behavioral/movement parameters," the system could generate ways to mitigate stress. Another approach which expanded the connection between mood, specifically anxiety and mouse usage, found that physiological indicators reflected strongly with physical action measured through mouse activities [8]. An example of understanding mental fatigue using both keyboard and mouse biometrics is another example of what can be learned from HCI [9]

Further work that sets the foundation for even more in-depth research in this field has developed advanced mouse tracking techniques [10]. These techniques employ the utilization of temporal/spatial analysis of mouse movements and actions to gain insights into human behavior that were not available in other more traditional systems. A framework and process are established for other researchers in this field, providing a common approach to simplify and unify later works utilizing the free software MouseTracker [11]. Hehman discusses the insights that can be gained using mouse trajectory data and how to analyze such data. By conducting straight forward trials, it is outlined how researchers can analyze various metrics, such as temporal motion, velocities, accelerations, spatial disorder, the differences between smooth and abrupt responses, and principal components.

Another recent work has advanced related topics using mouse tracking to study social categorization and self-control [12]. This work also examines the features of uncertainty and decision conflict of a user with a mouse. Stillman's work summarizes other results involving mouse trajectory analysis and how it is studied in two primary domains, social categorization, and self-control. Social categorization involves determining some subjects, such as determining if a person is a man or woman. Examining self-control, conflict in this domain resolves around a person's inner conflict in deciding what choice to make, such as choosing soda or water. Stillman reviews topics and metrics involved in mouse tracking, such as the area under the curve, maximum deviation, integration times, and response conflict. It is noted that the area under the curve and times a mouse crosses the x-axis, called x-flips, serve as explicit markers for confidence evaluation in mouse tracking.

Another examination [13] found mouse trajectory to follow the concept of approach-avoidance conflict, showing how indecision of choice affects mouse motion. This study discusses approach-avoidance conflict, meaning a conflict between making a choice that has both positive and negative associations, such as wanting to eat nachos, but aware of

the lactose intolerant consequences that would follow. It found that approach-avoidance in mouse motion shows differences in evaluating choices, making a choice, and then how one acts after deciding. They found that mouse motion was less complex than avoidance trajectories, meaning avoidance causes more conflict than choices that have inhibition associated with it.

2.2 Authentication and Biometrics

Different areas of research in HCI have considered mouse and keyboard dynamics to authenticate users. Authentication is the process in which a user's stated identity is confirmed. A demonstration of this worked with keyboard and mouse dynamics to detect the user's emotional state and identify them using a KNN classifier [14]. The use of unexpected questions to test identity against a built-in profile is another approach taken by researchers [3]. This idea of building a user profile while using a device and then using mouse dynamics to re-authenticate was proposed in earlier research [15].

Various other works have used mouse and keyboard biometric dynamics as a means for authentication and identification [16]–[20]. Additionally, using these methods enables the system to continually authenticate a user while using a device, preventing the case when a user steps away. A different user starts operating the device [21].

2.3 Metacognition Choice Confidence

Mental cognitive processes and metacognition, or the observation of our thoughts, such as knowing how confident or sure we are of something [22], have a measurable impact on external physiological behavior [23]. Prior work has identified processes in the human brain decision making and expressing that choice with a measure of confidence [24]. This work has measured neural activity through fMRI while a decision is being made [25]. Age, however, has been found to influence the strengths of a person's decision confidence [26]. An examination by [27] shows a high correlation between the cognitive mechanisms of making a choice and the relation it has to the motion to implement that choice. There is also a connection between decision and eye movement [28]. What is more, a person could measure their confidence externally. Decision confidence can be accurately stated by the person who decided due to the meta-cognitive nature of making decisions [29], [30].

2.4 Lie Detection

There has been work done to examine a person's honesty using various digital metrics such as typing speed and mouse movement. A study aimed to perform lie detection through mouse dynamics found that falsified information about identity could be detected using mouse motion [31]. In another work [32], a respondent's honesty was determined by developing a machine learning classifier that analyzed the mean time between keystrokes on a keyboard. Composing various questions that decided what honest and dishonest responses looked like, dishonest responses showed a noticeable difference in timing. This was due to the mechanisms required to tell a lie, which involves suppressing the honest answer, developing a lie, and then the act of expressing the lie. Because the lies were not a known fact, it took increased time to execute the physical action to type out the dishonest answer.

Similar patterns of mouse motion were leveraged in another study [33] which determined the legitimacy of a user's response on online insurance forms. Again, a pattern was seen where a person's physical motion was different when it came to answering honestly and dishonestly, with different velocities and trajectories. Similarly, a model was developed that was able to determine the honesty of a response accurately. The detection of hidden threats that have infiltrated an organization using hidden tests [34] was able to detect users who had antagonistic intent. Using "concealed information tests," they could administer online tests that could identify users who were guilty of working against their organization. This demonstrates the value of such tools, which identify various features of a person online, such as an understanding of human behavior, improvements in operational security, and the impact it holds on individuals and on what online activities can be leveraged against them.

2.5 Recurrent Neural Networks

Deep neural network models have been shown to accomplish a wide variety of regression and classification tasks using machine learning, such as big data processing [35], object detection for autonomous vehicles [36], and medical imaging analysis [37]. A neural network works using an array of nodes commonly referred to as 'neurons' with weighted connections to an input array; each node outputting a value in response to the input value. A neural network becomes deep when multiple layers of neural networks are stacked on top of each other. One method of training a model is supervised learning, which starts by feeding a sample of data representative of a problem, comparing the model's output against the sample's

known target response, and then performing gradient descent. Common neural network architectures, like perceptrons and convolutional neural networks, are often used on fixed dimensional inputs and outputs, such as images or tabular row entries. This, however, lacks flexibility when working with data of variable sizes, such as coordinate pair sequences or audio tracks. Because going a single timestep at a time would lose significant context, one method of handling sequential data is to create segment 'windows' of a fixed width as an input sample. However, the drawback of this approach is that a network can miss larger contextual patterns from previous windows [38].

Recurrent neural networks (RNNs), on the other hand, are a type of neural network that has the output fed back into itself. Training an RNN, the network is provided timesteps of data one at a time, producing a result that is then compared to the next timestep of the input and then fed back into the network and the next timestep. This allows a model to maintain a chain of context and allows the network to retain information from previous steps forward through time, enabling it to understand the data better. This is known as a distributed hidden state, which can be updated through non-linear dynamics while the model processes an input sequence. These properties make RNNs extremely powerful, so much so that they can be Turing complete, given enough resources [39]. They can be applied to tasks that rely on contextual structure, such as human speech recognition and translation [40].

Because of the structure of RNNs, there are several ways to use them to accommodate different functions. The first is the one-to-one, essentially a perceptron, where a single input is fed into the network, and a single output is taken. This behaves practically like a perceptron, as no next timesteps are used. The following is one-to-many, where the input can be thought of as a 'seed' to the model, which returns an output that is also used as input to the next output. The seed value sets the model's state in this orientation, which will then progress forward according to its learned pattern. Then the reverse, many-to-one, in which a sequence is used as input, ignoring the returned outputs except for the last. Here, the model sees a complete sample and will make a prediction influenced by the entire sequence. There are then two ways to utilize many-to-many. The first is like many-to-one, except all outputs are returned and not just used internally. With this, the network creates a new sequence that will likely resemble the input closely. The final method is to predict what a continuation of the input would look like, instead of where the many-to-one would end, instead of having the model continue returning outputs based on the entire input sequence [38].

An issue that occurs with normal RNNs is that a network will not retain enough information in their hidden state to retain long, intricate patterns over a long period. A modification of the RNN known as the long-short term memory (LSTM) introduces memory to a node, along with four gates that regulate what and when to remember and when to forget the values stored in memory and what to return. These gates helps the network retain information by giving it explicit values, which can apply context to later states. It also serves to alleviate a common issue with RNNs of exploding and vanishing gradients, which will cause the weights of a model to grow out of bounds, causing returned values to skyrocket or shrink to zero, which the returned value also goes to zero. In both cases, the model will no longer return meaningful responses [39].

CHAPTER 3:
Methodology

Using data collected from previous work in HCI, an LSTM neural network is trained to make predictions as to the most likely trajectory from the first few steps of sequences. A description of the original datasets and their goals and how it is formatted. The procedure that was used to transform the data into the training, validation, and testing datasets. Next, how the LSTM model was built and the configuration used during training. Finally, a discussion of the LSTM's predictive power to measure confidence as a comparison of predicted and real mouse motion.

3.1 MouseTracker Data: Description and Processing

Mouse data is generated by a pointing device, generally a mouse, at various rates, sensitivities, and accuracy. Most mice have a resolution known as dots per inch (DPI), ranging from 400 to 1600 and above. This resolution determines a mouse's sensitivity, as the higher resolution allows the mouse to create tighter motion tracking. When moving, a mouse produces position deltas at their refresh rate, which is then polled by the operating system, translating these changes into motion. A tool that standardizes mouse data for HCI research involving mouse dynamics is MouseTracker, created by Jon Freeman [11]. MouseTracker enables researchers to design experiments to be administered to participants through a series of trials. Experiments typically consist of a prompt at the bottom of the test screen, with a box above representing the testing field. Then, at the lower center of this field is a box where the mouse pointer will start for every trial, where the user will respond to the prompt by moving the mouse's pointer to click on one of two choices at the top left and top right regions of the testing field. From the start to the end of the trial, the mouse's position is recorded on an X- and Y-coordinated plane polling rate of the operating system, which on Windows is between 15 to 16 milliseconds. MouseTracker can compute metrics with the recorded data, such as mean trajectory angle, the area under the curve (AUC), and along the X- and Y-axis. This tool allows researchers to construct trials that examine specific criteria involving lie, confidence, and other cognitive processes. However, in this work, a more generic method of determining user confidence will be explored, disconnected from specialized software and

only using the mouse's trajectory data itself. Freeman notes that MouseTracker was developed to accommodate general hardware evenly but notes that different pointing devices may record motion at different sensitivities and be accounted for by researchers before testing.

This software helps alleviate issues commonly found in mouse trajectory analysis. Mouse driver software differs depending on system configuration and hardware, where mouse position is polled at different rates depending on the mouse's motion. When at rest, a computer will poll the mouse at lower frequencies, then ramp up to a constant rate during movement. This variance in the recording can cause issues getting accurate mouse position data as the timing will be less accurate early on than later in the tracking. With MouseTracker, the mouse's position is polled at a relatively stable 15 to 16 milliseconds, resulting in much more consistent monitoring.

Data was sourced from four studies on intertemporal decision making and temporal discounting, which utilized mouse dynamics using MouseTracker to conduct data collection and analysis. The data is available unrestricted from the online databases Figshare.com and Mendeley.com. The first study [41] utilized resting state functional connectivity MRI along with mouse trajectories, which found a connection between specific neural pathways and temporal discounting. The second [42] examined gambling behavior regarding choice, seeing that pathological gamblers prefer smaller immediate rewards to larger ones in the future. Third, a study [43] looking at making choices in episodic future thinking when positive and negative qualifiers are included in questions. Lastly, a look into cognitive dynamics to religious belief [44], which saw a relation between religious commitment to temporal discounting. In each of these works, confidence was evaluated by computing the AUC between the mouse trajectory and an ideal straight trajectory between the start and finish positions. The max deviation (MD), which was the furthest the cursor moved away from the perfect course.

The data is saved as plaintext files with an extension of .mt for each subject in a trial, containing six types of track segments. Files are structured like a .csv file, as commas delimit each column until the last column, delimiting a sequence's timesteps. Each section has a header describing what that section relates to, and the row under describes the contents of the following row. Each row of a segment represents that section's track for a specific trial and contains several fields. The identification fields are subject ID, which uniquely identifies

the subject in a research project, a trial order number, and a whole number to identify a subject's trial. Then the temporal data: start times and the data track which contains the time or coordinate data of floating point or integer values. Other values not used are trial identifiers for stimulation, response options, chosen response number, condition, unused blank error values, and comments. Each trial is joined across segments based on the subject ID and trial number, though real-time is used to identify a trial further uniquely. The first segment is for space-scaled and time-normalized right-remapped data, which is deprecated and discarded as it contains no tracking data. The next two segments are for space scaled Y- and X-coordinates, which are normalized according to MouseTracker's testing field. The X-track ranges from 0.0 to 1.5, and the Y-track ranges from 0.0 to 1.0, both with an accuracy of four decimal places. This scaling is due to MouseTracker's testing field, having a size ratio of 2:3, designed to accommodate most modern monitors' shape. The next two segments are the raw coordinate data from each trial, with the X-/Y-coordinates being the pixel values on the screen. Values range depending on the screen size and resolution that a trial was conducted on, X-coordinates ranging from 0-1500 and Y-coordinates between 0-900. The final segment contains the increasing time track, representing each step's timestamp since the trial's start, expressed in milliseconds.

3.2 Processing

We wrote a script to open a collection of .mt files and export a .csv file. Each file is read from a data directory recursively, which reads line by line each track, determining what type of track is being read based on each segment's header. Trials are organized in a dictionary according to an identifier constructed from the subject id, trial number, and trial start time and hold the tracks stored in lists for each entry, and example of which is seen in Table 3.1 for the original data and Table 3.2 for the resultant dataframe. At this time, Y-tracks are flipped so that the origin point goes from the top left, as it is read on MouseTracker, to the bottom left, to make transitioning to other coordinate orientations more convenient. Once all .mt files have been read, the whole dictionary is saved in a .csv file to be loaded during training. Figures 3.1 and 3.2 give examples of what trajectories look like and how they are visualized.

Table 3.1. MouseTracker Data Sample.

.mt data, raw X-position (in pixels)				
subjID	order	init time	real time	Timestamps
1.1	1	1609	10.05.19	0,33,. . . 50
1.1	2	313	10.05.23	2,10, . . . 45
1.2	3	390	05.05.26	0,2,. . . 100
.mt data, raw time (in ms)				
subjID	order	init time	real time	Timestamps
1.1	1	1609	10.05.19	0,15,. . . ,730.
1.1	2	313	10.05.23	0,16, . . . 403
1.2	3	390	05.05.26	0,15,. . . 830

Table 3.2. Converted MouseTracker Sample to Dataframe Format.

subjID	order	init time	real time	X-Position	Time
1.1	1	1609	10.05.19	0	0
1.1	1	1609	10.05.19	33	15
. . .	. . .	. . .	. . .	. . .	. . .
1.1	1	1609	10.05.19	50	730
1.1	2	313	10.05.23	2	0
1.1	2	313	10.05.23	10	16
. . .	. . .	. . .	. . .	. . .	. . .
1.1	2	313	10.05.23	45	403
1.2	1	390	05.05.26	0	0
1.2	1	390	05.05.26	2	15
. . .	. . .	. . .	. . .	. . .	. . .
1.2	1	390	05.05.26	100	830

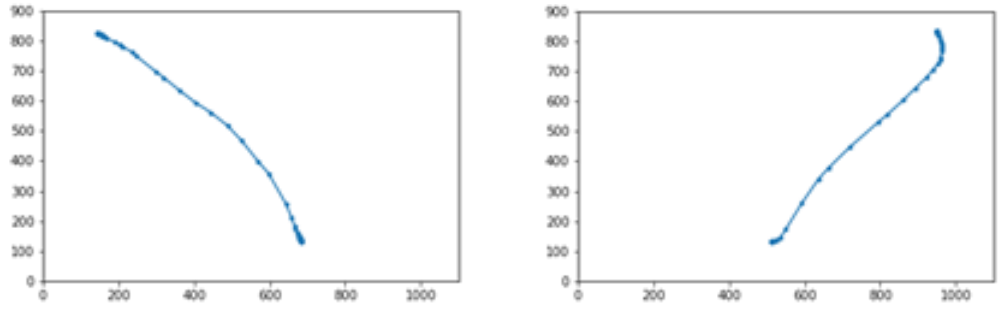


Figure 3.1. Example Simple Motions. Motions exported from MouseTracker files which exhibit simple trajectories.

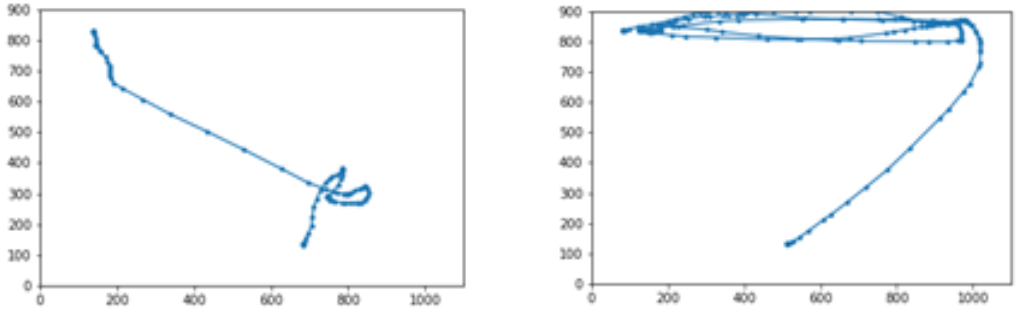


Figure 3.2. Example Simple Motions. Motions are exported from MouseTracker files, which exhibit complex, longer trajectories. These trajectories are demonstrative of a user unsure of what choice to make, moving around the field as they come to a decision

After processing from the .mt format, the data is loaded from the previously constructed .csv. The data is stored as a Pandas dataframe, selecting from the file the unique trial ID, raw X, Y, and elapsed time tracks of each trial. Here, the coordinates are divided by the absolute largest space coordinate tracks' absolute largest value to normalize them in a -1.0 to 1.0 range. Next, tracks representing the delta of the X, Y, and elapsed time tracks, first computing position deltas, collapsing any steps which saw no change in both X- and Y-coordinates, and then the time delta is computed. This computation shortens the time-series into immediate consecutive transitions, the pauses between mouse movements being reflected in longer time deltas, like how mouse data is normally produced. Distance between points, X/Y velocities, the speed of motion, and the angle between points are then calculated to provide deeper

training context. The final tracks used to train the model are normalized X-/Y-coordinates, the change in the time between timesteps, the change of X and Y, the velocity of X and Y, the speed, and angle. Equations 3.1 - 3.8 were used to calculate features.

Delta X-/Y-Position and Delta Time:

$$\Delta x_n = x_n - x_{n-1} \tag{3.1}$$

$$\Delta y_n = y_n - y_{n-1} \tag{3.2}$$

$$\Delta t_n = t_n - t_{n-1} \tag{3.3}$$

Distance:

$$d_n = \sqrt{\Delta x_n{}^2 + \Delta y_n{}^2} \tag{3.4}$$

Velocity:

$$v_{xn} = \frac{\Delta x_n}{\Delta t_n} \tag{3.5}$$

$$v_{yn} = \frac{\Delta y_n}{\Delta t_n} \tag{3.6}$$

Speed:

$$s_n = \frac{d_n}{\Delta t_n} \tag{3.7}$$

Angle:

$$a_n = \tan^{-1}\left(\frac{v_{yn}}{v_{xn}}\right)/360 \tag{3.8}$$

Further modification stemming from LSTM layers through Keras requires that all sequences be of equivalent shape across all batches during training to take advantage of hardware optimizations implemented through Tensorflow; they must be post-padded to equal length, typically with zeros.

The data will go through a data augmentation to make it suitable for training using Tensorflow with Keras during training. Trials are selected randomly from a batch, replaced with a copy flipped around the starting point. This is done for two reasons, first is for left- and right-handed agnosticism in the training data, as the original data did not have this information included. Second, this doubles the amount of data seen during the model training. Figure 3.3 gives an example of a flipped trajectory.

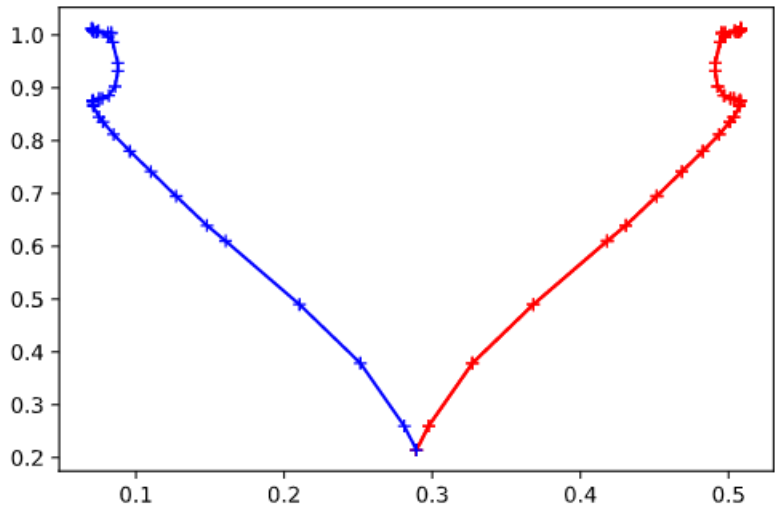


Figure 3.3. Example Flipped Motion. Motions which are flipped to provide left- and right-hand agnosticism and increase training space. Red is the original motion, blue is the flipped sequence centered around the starting point

Additional changes are done to take advantage of Keras and Tensorflow training optimizations for LSTMs when all sequence's shape match for the entire dataset. If the sequences' shapes do not match, they must be post-padded and masked. Because of the variable length of a trial sequence, trials were post-padded with zeros to a length equal to the dataset's longest track. A masking step then prevents the loss function from computing these timesteps' errors, preventing them from being learned by the model.

3.3 Model Design

Using the Keras machine learning API, the model is built in stacked layers. Input to the model is handled by a masking layer, telling the training engine not to use a specific value for

training, in this case, zero, so it does not influence weights. Kera's requires the input shape to the masking layer be of the shape batch size, maximum sequence length, and the number of features, as was explained earlier. The masking input layer outputs into two stacked LSTM layers, each with 1024 nodes. NVIDIA's CUDA GPU acceleration optimization requires the following: hyperbolic tangent function for activation; recurrent activation uses a sigmoid function; no recurrent drop out is used; weights are initialized using the Glorot uniform initialization, with the recurrent initializer being orthogonal. Each LSTM layer is set to return a sequence only to return the next predicted step by default. The models are also initialized statefully during inference, allowing more flexibility during prediction as the hidden state will propagate to the next prediction step and is reset manually. Additionally, the output is a dense layer with the number of predicted features as the output with no activation or bias. This final layer serves to interpret the output from the last LSTM layer into the features that are being predicted.

3.4 Training and Evaluation

The model is evaluated during training using an ADAM optimizer on the validation loss, which is the mean square error. Additional metrics used are mean absolute error, mean squared logarithmic error, for all features, and mean squared, mean absolute, and mean squared logarithmic error only evaluate on the predicted X, Y, and time delta. These give a clearer picture of the model's performance on the whole feature range and the original data used to construct the features. This divided two-layer evaluation allows for a close-up view of performance and allows a better understanding of the model is predicting well. It is possible that the model does well on predicting derived features such as coordinate deltas, angles, and velocities, but less well on position and timestep delta. These metrics will focus on evaluating the differences in performance between original features and derived.

During training, the model is saved after every epoch if the validation loss has shown improvement over the previous epoch. The training will continue for another thirty epochs after the last best model, after which it will stop training.

3.5 Application of LSTM for Confidence

The use of RNNs fits well with the sequential nature of the mouse-tracking data. An LSTM architecture was chosen because of its resistance to exploding/shrinking gradients on longer sequences. The model is designed contrary to common practice to capture the less complicated mouse data patterns and instead be more effective at predicting more direct, less complicated motions. The intuition here is that direct movement is a confident motion, while a more complex mouse motion involves conflict between the available options. This is reinforced by previously mentioned work involving choice confidence and decision making [41]–[44], as well as others [45].

In prediction, two methods are used to evaluate confidence by comparing the predicted and real outputs. 1) Compute a many-to-many inference where the output is the prediction at every step informed from every previous step evaluated. Comparing the predicted sequence against the ground truth reveals how well the model has learned to interpret the data in a general sense. 2) Seed a model with several steps from a trial and then recurrently predict using the previous result. Predictions are made until the output's length is the same as the seeding sequence. The neural network will predict the most likely outcome based on what it learned. Exploring the predictions this way allows evaluation of confidence based on how sporadic the motion is, with smooth motion having lower error than the movement that jumps around in trajectory. That is, when a course is straight-forward, the most likely result will mimic the most confidence, and vice versa. When a trajectory is complex, the predicted motion also becomes confused, reflecting indecisive motion [45]. The original metrics of confidence are then measured and compared against the prediction error of each trial. A low error rate would correlate with a high confidence metric and a high error rate with a lesser confidence metric. With these metrics, the test trails will be evaluated to determine a proxy for confidence prediction. This method allows for a generic view of confidence prediction, wherein other studies, application-specific models must be used depending on the area of cognition being evaluated in confidence. Here, a mouse's motion can be used to assess a user's confidence based purely on the direct action observed, disconnected from questions being asked. This better reflects real-world applications, as the context may be too nebulous to be understood, or the stimuli of a mouse's motion not known.

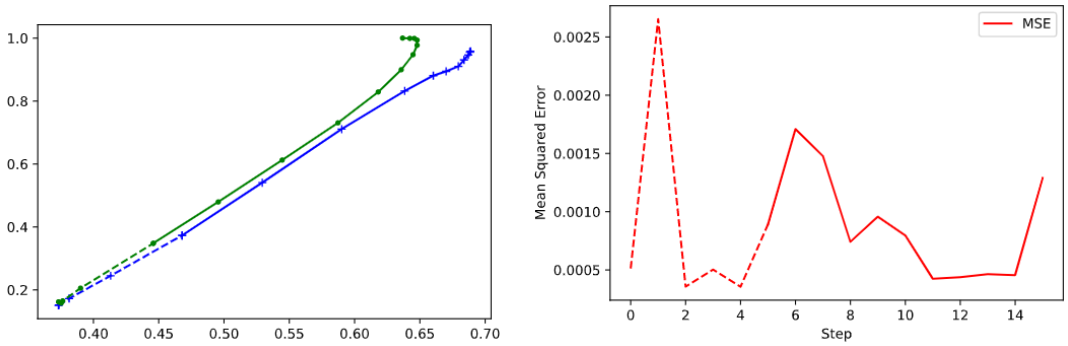
THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

Examining the data reveals the complexity of mouse motion and reveals several strengths and shortcomings of the model's ability to make predictions. This approach has not been well explored, but its novelty may hold potential in confidence measurement. It is shown that the model can produce a lower error on confident movements and a higher error on less confident movements. However, there are cases where the error will be lower than confidence metrics, indicating the model reproducing indecisive motions.

4.1 Evaluation Metrics

Predictions are generated from a model in two ways. First, evaluating every sequence in the test set 8736 trials from 533 individuals to reproduce the MSE used during training by first taking a sequence of length n and copying them into an X stimulus from $0...n-1$ and Y-hat target from $1...n$. X is used as input, which the model iterates over every timestep t from $0...n-1$, producing a prediction for $t+1$. The result is a sequence of predictions from $1...n$, which are then used to compute the MSE with the prediction and Y. The next prediction method is to supply 'seed' sub-sequences of sequence X of length $s \in \{5, 10, 30, 50, 75, 100\}$ of each sequence. If a sequence length $n \leq s$, then a sequence is used as a seed up to $s < n$. This produces a predicted sequence of length s. The model then repeatedly takes the last predicted timestep, appending the predicted timestep to the sequence until the sequence is length n. The predicted sequence and sequence Y-hat are again used to compute the MSE, coordinate MSE, and forward MSE, the error for all steps after the last seed timestep. An example of the seeding and prediction is given in Figure 4.1.



(a) Stimulus and Prediction

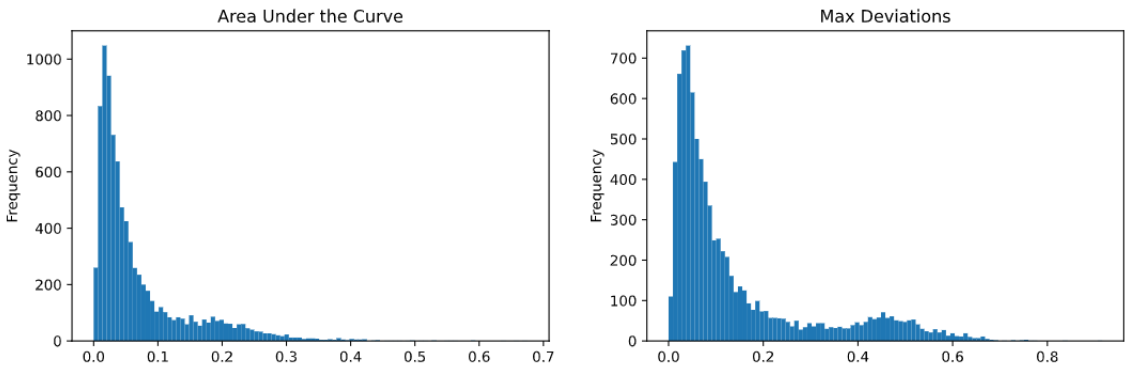
(b) MSE of Each Feature at Timestep

Figure 4.1. Example Prediction. (a) In blue, the ground truth created by the subject, in green, the line predicted by the model. The dashed lines represent before steps seen, solid after. The prediction has a MSE of 0.00086 and a forward prediction MSE of 0.00097. (b) At each time step, the MSE is computed of all features, dashed being during the seed phase, and sold during forward prediction.

4.2 Test Data Description

Confidence evaluations were done by comparing the metrics of the ground truth trajectories against the prediction error. The AUC and the MD were calculated on the original trajectories to produce a metric to measure confidence. With caveats that these studies use these metrics in specific manners to achieve more nuanced results. Here, AUC and MD are used to approximate a subject's confidence from their mouse trajectories.

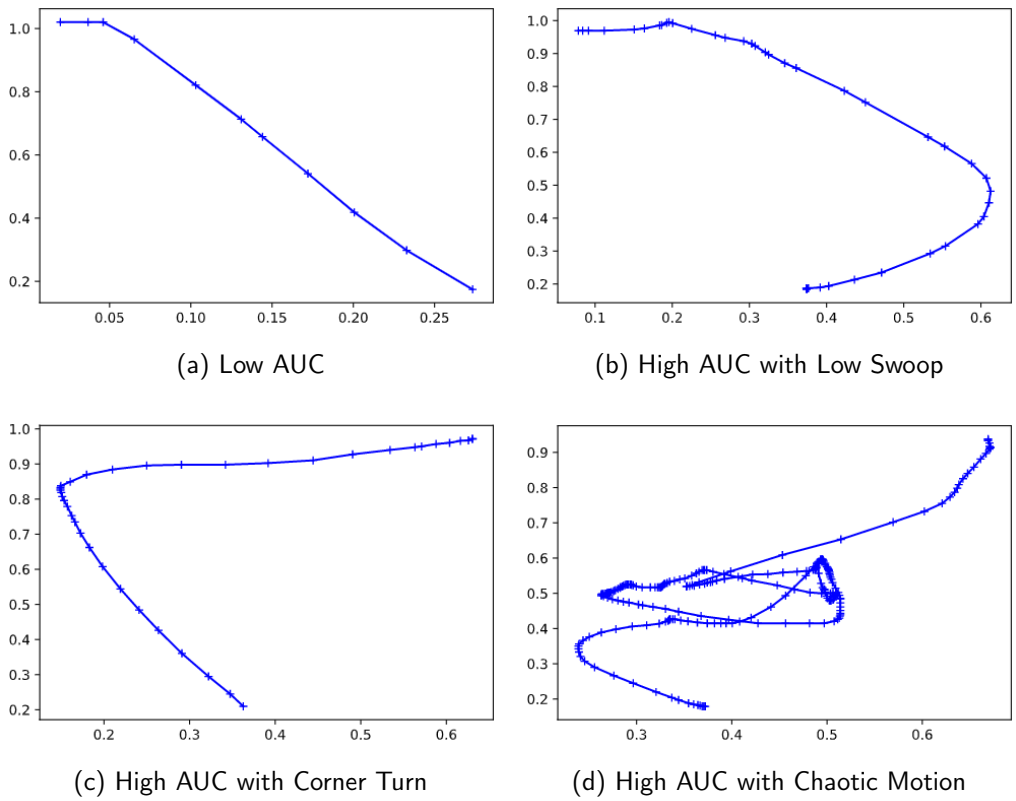
The trial data features two primary features, which were used to measure the confidence of a trial. A total of 8736 trials from 533 subjects with a mean count of 16.4 trials per subject, with a std of 5.2, and trial lengths had a mean of 55.7 steps with a std of 36.7. The AUC and MD of each trial were measured, with the AUC being 0.071, with a std of 0.078, which is used as the primary measurement of confidence of a trial. At the same time, MD is considered secondary, as it has some more significant variance in the distribution of 0.144 and a std of 0.156. The histograms in Figures 4.2 show the distribution of AUC versus MD scores.



(a) AUC (b) MD

Figure 4.2. Histograms of Evaluation Metrics. Histograms of the measured metrics used on the original curves to measure confidence. Figure (a) is of the area under the curve, and Figure (b) is of the max deviation.

A better understanding of the model's performance can be gained by examining a selection of predictions. Typical behavior of a trajectory generally exhibits several shapes. One is a straight motion from start to end position, with little deviation, seen in Figure 4.3a. These types of curves are typical of highly confident choices, as there is little conflict from the subject regarding what choice they are going to make [13]. There is more variance in trials with higher AUC and MD, which is expected as the participant will have more significant conflict in their choice. These curves fall into three general categories: a low swoop at the beginning of the motion, as shown in Figure 4.3b. The second type of curve (Figure 4.3c) consists of the mouse moving towards one choice. After reaching or coming close to it, it changes direction and moves horizontally towards the second choice, sometimes flipping between the two options. The most chaotic type of motion (Figure 4.3d) is one that consists of random movements sporadically through the test space.



(a) Low AUC

(b) High AUC with Low Swoop

(c) High AUC with Corner Turn

(d) High AUC with Chaotic Motion

Figure 4.3. Examples of the types of curves seen

One final situation which helps explain the aberrations in the MD is caused when a subject moves the mouse diagonally from the ideal trajectory between the start and finish and then return to travel closer to the perfect trajectory, which results in not increasing the AUC as significantly as the MD, such as the one in Figure 4.4. While there are potentially other causes of this discrepancy, this was the most common one found. This quick reversal displays a conflicted choice being made. Still, once a trajectory reaches one of the two positions, it is likely to start moving towards the other choice because of how the model's behavior.

To gain better insights into the model's performance in confidence prediction, the prediction exposes high and low confidence measurement, exploring an intersection between the greater and lesser values of AUC and MD against high and low MSE, coordinate MSE, and forward MSE. The lower 15th quantile of both measurements has an AUC mean of 0.0099 with a std of 0.0035, and an MD with a mean of 0.0208 and a std of 0.0090. These sequences

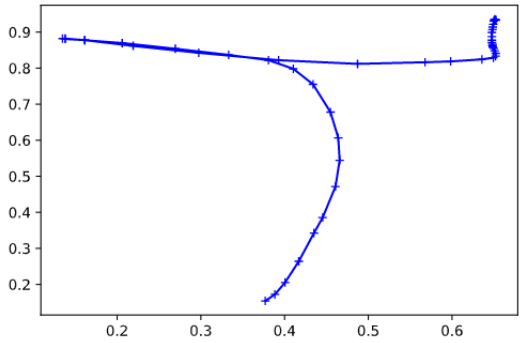


Figure 4.4. Example of Higher MD with Lower AUC

are mostly direct trajectories from the start and finish, typical of confident choices having a little conflict. On the other side of the spectrum of mouse-motion are far more complex motions, with an AUC of 0.2287 with a std of 0.0712 and MD of 0.4575 and std of 0.1057. Another metric that does not relate well to model performance is flipping along the X-axis. The model tends to 'jitter' at times, causing increased x flips while not being present in the predicted trajectory.

4.3 Model Performance

The model performed 28505 total predictions, with the breakdown of the performance at each seed step listed in the Table 4.1. As seed steps increase, MSE, coordinate MSE, linearly scaled MSE, and forward MSE trend downwards as the model is given more generous context to a trajectory. It can be said of the coordinate MSE that the model has difficulty predicting future coordinate positions. Metrics such as velocity and time delta have smaller values and less variance, making them easier to predict accurately. This imbalance is likely due to the position errors being of greater magnitude. For example, the delta time value has a mean of 0.00016 with a std of 0.00008, which the model would predict with a MSE of -0.00057 with a std of 0.000247, while the Y-position has a mean of 0.33862 with a std of 0.16790. in this case, it becomes clear that the positional data are a sizable part of the error. In the time delta, this vast space is due to no small time delta present in the dataset but was not present in the testing data.

Table 4.1. Error Statistics for Entire Testing Set

		MSE		Coordinate MSE	
	count	mean	std	mean	std
-1	8736.0	0.001688	0.001070	0.001774	0.003073
5	8736.0	0.016089	0.012033	0.055717	0.050110
10	8608.0	0.013233	0.011086	0.044950	0.045779
30	7040.0	0.007669	0.007999	0.024125	0.032635
50	4096.0	0.006246	0.006799	0.019110	0.027643
75	1888.0	0.005754	0.006329	0.017460	0.025783
100	896.0	0.005461	0.005753	0.016615	0.023439

		Forward MSE		Forward Prediction Length	
	count	mean	std	mean	std
5	8736.0	0.017564	0.012894	50.704899	36.673930
10	8608.0	0.015773	0.012638	46.442728	36.438753
30	7040.0	0.013242	0.014310	34.200994	35.856109
50	4096.0	0.012829	0.011919	32.750000	36.901784
75	1888.0	0.013458	0.012221	34.931144	39.008166
100	896.0	0.014070	0.011700	37.876116	40.760689

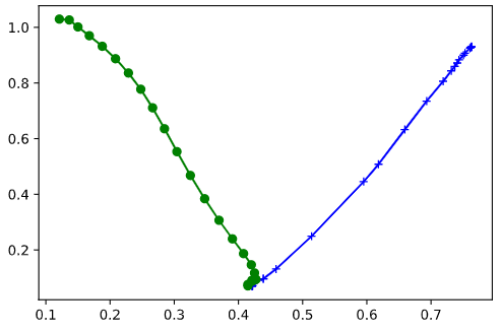
We see the values of the forward prediction error in Table 4.1, the error of the prediction after seeding the model against the rest of the sequence, is the primary focus on the model's evaluation of confidence. A correlation can be seen in the Table 4.2, especially at seed step 10 with a Pearson coefficient of about 0.248. This correlation leaves room for improvement but suggests more than can be learned from this method to produce confidence measurements from LSTM neural networks.

Table 4.2. Correlation Between Error and AUC/MD By Step

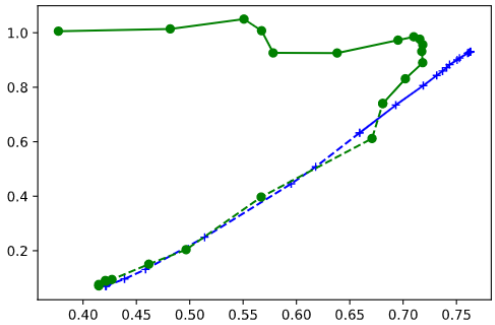
Step	-1	5	10	30	50	75	100
	MSE						
AUC	0.079571	0.197810	0.265611	0.350096	0.354358	0.293618	0.248201
MD	0.040964	0.187642	0.251060	0.329269	0.323190	0.250268	0.182046
	Coordinate MSE						
AUC	-0.119118	0.201927	0.267292	0.342016	0.355866	0.292060	0.246376
MD	-0.172710	0.190196	0.250521	0.319823	0.323445	0.249773	0.180873
	Forward MSE						
AUC		0.185968	0.248939	0.067104	-0.015060	0.012102	0.044923
MD		0.174814	0.232700	0.071438	-0.006155	0.020337	0.063397

The network seed of ten producing the highest result is also due to most trials' mean length. At five steps, the model will not have been exposed to enough data to know which direction the mouse will be moving, as in many cases the mouse has only made small shifts near the start, which the model will either also idle near the start position or choose a direction to move in. While this is indicative of indecisiveness, it contributes to the variance of the model's prediction error. It becomes more likely that the mouse has progressed towards an end goal at ten steps, and the model will predict closer to the expected output.

In Figure 4.5 the dashed blue and green lines are the seeding sequence for the ground truth and prediction, respectfully, while the solid green is the model's predicted motion from recurrent predictions. The solid blue is the unseen ground truth. This example also displays another intriguing feature of the model and the data. Because of the mouse's idling, the model will tend to move towards the opposite area, even after it has been given more context to the mouse's trajectory, as seen in Figure 4.5b. This result is typical, as the model tries to mimic the second type of indecisive curve discussed earlier, which moves first towards a choice, then towards the other.



(a) Example of Early Diversion Prediction.



(b) Example of Switchback Prediction.

Figure 4.5. Example Ground Truth and Prediction.

Breaking down the prediction and confidence metrics into four categories shows where the model does well and does not offer high or low confidence. First, there is low AUC/MD and low forward MSE, which can be confident answers that the model can predict accurately and read as a true positive. The second is the false-negative case, with a low AUC/MD but high forward MSE, the prediction and ground truth diverging significantly. The model interprets a confident curve as unconfident by misunderstanding what direction it will be moving. Then there is the true negative, with a high AUC/MD score and high forward MSE resultant from the model diverging from the ground truth because it was not clear where the trajectory would lead to because of unconfident motion. Last is the false negative. The model begins to predict future steps even though the curve has a high AUC/MD score but resulting in a low forward MSE diverging from the ground truth because it was not clear where the trajectory would lead to because of unconfident motion. Last is the false negative, where the model begins to predict future steps even though the curve has a high AUC/MD score but resulting in a low forward MSE.

Two case studies will be discussed for each category, displaying the real and predicted mouse trajectories to give a visual context and the squared error score at each time step. These case studies provide a clearer picture of the model's behavior and what it can predict. While there is a large variety of trajectory shapes, this breakdown will serve as a groundwork to better understand the qualities of LSTMs for motion prediction context and the squared error score at each time step. While there is a large variety of trajectory shapes, this breakdown will serve as a groundwork to better understand the qualities of LSTMs for motion prediction.

Correlations are drawn from comparing the ground truth AUC and MD against predictions metrics at each time step. These comparisons explain how accurately they relate, though they are not directly comparable because of the disproportional relationship. However, it does serve as a reinforcement when taken into consideration and visual exploration of predictions against ground truths. Forward MSE is also not computed for the full forecast, as there is only a single prediction at $n + 1$.

4.4 Accurately Measured Confident Trajectories

The first type of prediction is lower AUC and MD and a lower prediction error rate. These types of forecasts lacked complexity in their trajectories, traveling directly from start to finish. In this selection of trajectories, there were 1311 trials chosen from the test set, which were in the 15th quantile of AUC values, with an AUC mean of 0.00987 with a std of 0.00350, and MD mean 0.02082 and std of 0.00898, and the 15th quantile of forward MSE, listed in Table 4.3. Table 4.4 presents error rates at each seed timestep.

Table 4.3. Low AUC and Low MSE By Test Set

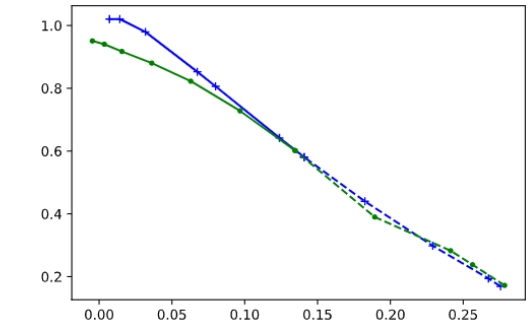
	MSE	Coordinate MSE	Forward MSE	Forward Prediction Length
Mean	0.001829	0.003129	0.002401	19.079365
Std.	0.000869	0.002439	0.000964	10.4946043

Table 4.4. Correlation Between Error and Metrics for Low AUC and Low MSE By Seed Step.

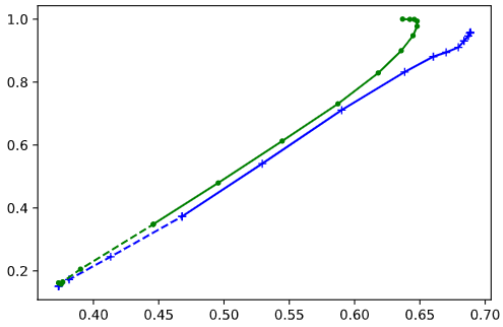
Step	-1	5	10	30	50	75	100
	MSE						
AUC	0.047653	-0.013772	-0.010123	-0.101958	0.149861	0.13486	0.222468
MD	-0.07503	-0.092901	-0.180198	-0.123433	-0.076921	0.034501	-0.409254
	Coordinate MSE						
AUC	0.169297	-0.40405	0.100097	0.015516	0.250253	0.213964	-0.036979
MD	0.067916	0.181471	-0.163342	0.041305	0.079943	0.144367	-0.064376
	Forward MSE						
AUC		-0.01624	0.047922	-0.035937	0.055223	0.374901	0.008872
MD		-0.064836	0.005762	0.068138	-0.05895	0.012712	-0.098395

featured in Table 4.4 is that the forward prediction reverses in the 30 and 50 seed steps. This reversal is likely related to more trials here being shorter, suggesting the model has an easier time predicting shorter trajectories. Once they get beyond a certain point, they begin to lose their ability to predict low AUC curves accurately. As shown in the next section, longer trajectories generally cause the model to predict more sporadic motions, which causes them to be featured here less.

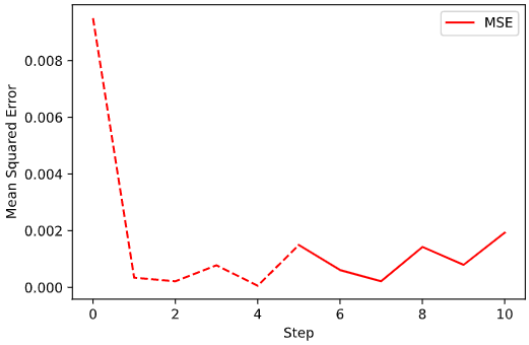
Figure 4.6 shows two predictions with a direct motion from start to finish. Figure 4.6a shows the model predicting the future outcome of the mouse's trajectory accurately, and Figure 4.6d displays another characteristic of these types of trials. First, the model will begin to pull towards the opposite answer. Additionally, the ground truth features a sudden slow down near the end of the trajectory, which the model reflects as well, with a closer bunching of points at the turning point.



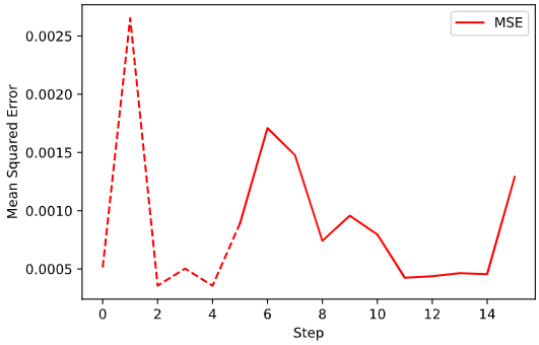
(a) Low AUC and Low MSE Example 1.



(b) Low AUC and Low MSE Example 2.



(c) Low AUC and Low MSE Example Error 1.



(d) Low AUC and Low MSE Example Error 2.

Figure 4.6. Example Low AUC and Low MSE Ground Truths And Predictions.

By looking further into the squared error at each time step as seen in Figures 4.6a and 4.6a, we can see what will be seen elsewhere. The primary features which contribute the most error are the X/Y coordinates, but so can the X and Y velocities. It is often the case that the speed and time delta error barely register, as they are scaled out against the other features. Another interesting observation is that errors in velocity prediction will sometimes be more significant than positional early on, but they begin to align as they progress. This change is potentially due to the 'warming up' of the model, gaining context to understand coming trajectories.

4.5 Confident Trajectories Predicted as Unconfident

From the previously mentioned quantile of original trials, the other end of the prediction error is selected, this time resulting in 253 trials of various seed step lengths. The lack

of trials is not surprising, as longer tracks that do not make for easy prediction will have a higher frequency of appearances with more seed steps, which can be seen in the mean forward predicted timesteps of about 53. Table 4.5 shows the error rates for the entire set, while Table 4.6 breaks down by seed step.

Table 4.5. Low AUC and High MSE By Test Set

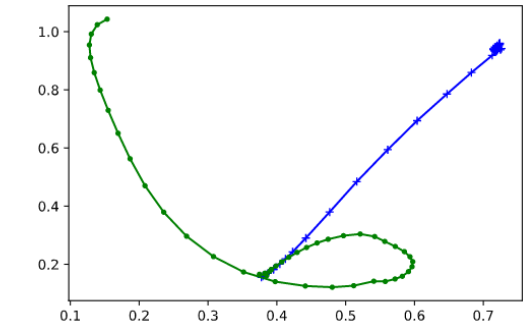
	MSE	Coordinate MSE	Forward MSE	Forward Prediction Length
mean	0.032277	0.122910	0.039735	52.936759
std	0.009358	0.038755	0.011080	23.274329

Table 4.6. Correlation Between Error and Metrics for Low AUC and High MSE By Seed Step

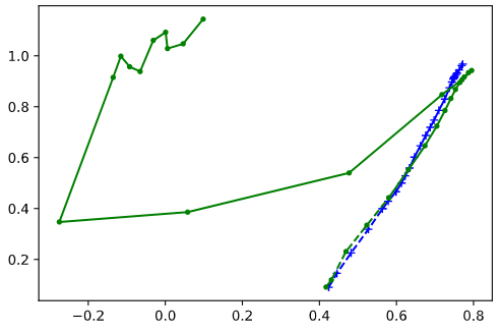
Step	-1	5	10	30	50	75	100
	MSE						
AUC	0.160695	Nan	0.484315	0.002496	0.246178	-0.041332	0.292143
MD	-0.006606	Nan	0.183553	0.091817	-0.201277	-0.013578	0.213027
	Coordinate MSE						
AUC	0.351643	Nan	-0.222984	-0.034092	0.00295	-0.206382	0.484061
MD	-0.084523	Nan	0.921947	0.140538	0.154078	0.024477	-0.114694
	Forward MSE						
AUC		Nan	0.979783	0.073348	0.218209	-0.065573	-0.106784
MD		Nan	-0.516619	-0.164952	-0.216881	-0.154557	-0.077884

Both ground truth paths in Figures 4.7a and 4.7b feature direct motion from start to finish, but here a key difference is that they have more steps, either along the way from start to finish or clumped together at either end. In Figure 4.7b, the model does predict the motion very well. Still, after the 15th step, predictions begin to jump around, uncertain of where to go next because of the cluster of small movements collected near the end of the motion. The subject moves slightly for some time before completing their choice. Another feature of this motion is sharp corners with long jumps between them far off the test area, though it

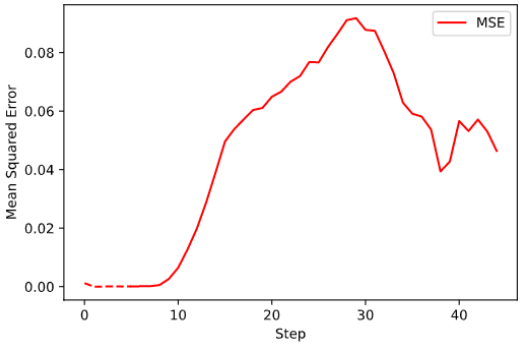
is unclear what might have caused these. These jumps contrast with Figure 4.7a trajectory's motion, which diverges early from the path suggested by the seed steps, flowing in a low swoop towards the opposite answer with no apparent cause. The difference here is that more time is spent near the beginning of the motion, confusing the model's future prediction, which is continued for so long because of the second grouping of points at the other end of the movement flowing in a large loop towards the opposite answer. The difference here is that more time is spent near the beginning of the motion, confusing the model's future prediction, which is continued for so long because of the second grouping of points at the other end of the movement.



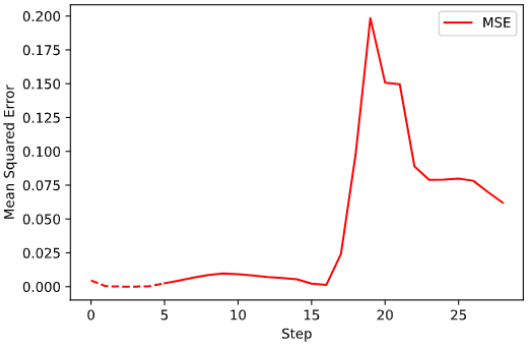
(a) Low AUC and High MSE 1



(b) Low AUC and High MSE 2



(c) MSE. Step: 5



(d) MSE. Step: 5

Figure 4.7. Example Low AUC and High MSE Ground Truths And Predictions.

4.6 Accurately Measured Conflicted Trajectories

These trajectories were selected from the test set by taking the top 15th quantile of AUC for another total of 1311 ground truth trials. These have an AUC mean of 0.22869 and std of 0.07128, and MD mean of 0.45746 with a std of 0.10569. As seen before with the earlier section, as the lengths of sequences increase, so does the potential for longer sequences to be included, meaning a total of 963 trials predicted over.

The general error for high AUC and high MSE is seen in Table 4.7. A critical difference between these types of trajectories is that MD negatively correlates to every error metric with a few exceptions. The cause of this is that the MSE can grow much larger than the MD, as the trial space bounds it. It is also evident by comparing the MD correlation with the MSE against the forward MSE at the 100th seed step. A sequence that follows a high AUC trajectory will have a higher MD than MSE, but excluding the seed phase error will result in a lower MSE in general and reduce the negative correlation. This is seen in Table 4.8.

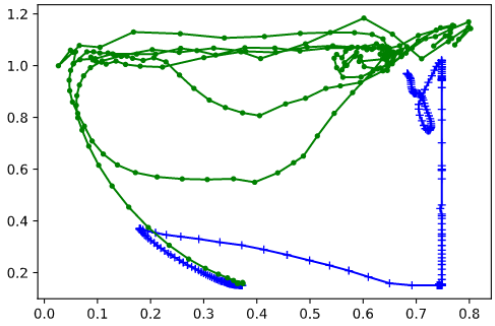
	MSE	Coordinate MSE	Forward MSE	Forward Prediction Length
mean	0.030420	0.119270	0.038317	108.759086
std	0.009782	0.041337	0.009317	60.923490

Table 4.7. High AUC and High MSE By Test Set

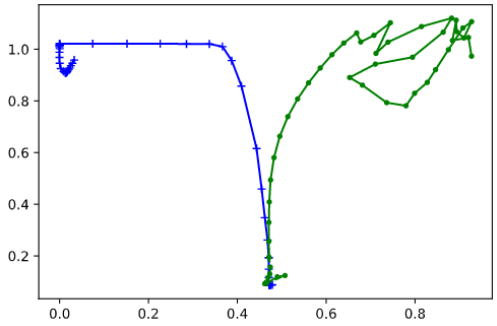
Table 4.8. Correlation Between Error and Metrics for High AUC and High MSE By Seed Step

Step	-1	5	10	30	50	75	100
	MSE						
AUC	0.132886	Nan	Nan	-0.63301	0.094296	-0.07213	0.117161
MD	0.029479	Nan	Nan	0.180156	0.174267	0.017177	-0.0487
	Coordinate MSE						
AUC	0.186293	Nan	Nan	-0.475522	-0.042991	0.082832	-0.059777
MD	0.125257	Nan	Nan	0.167031	0.301726	0.206291	0.081397
	Forward MSE						
AUC		Nan	Nan	-0.287671	0.097539	-0.160869	-0.16032
MD		Nan	Nan	0.103966	-0.08545	0.008321	-0.197074

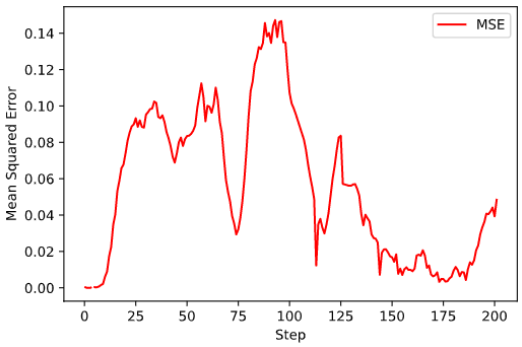
Figure 4.8 shows examples of the curves that show the most outstanding feature of conflicted choice. They have higher AUC, and because of their longer sequence length, they also cause the model to be unsure of future motion. The motion on the left shows the prediction following the initial direction and curvature of movement closely until reaching the typical endpoint at the top left. The model moves back and forth in smooth motions, which is typical of longer sequences, though not this one. This contrasts with the other trajectory, where the primary error comes from the X-coordinate moving directly opposite the ground truth. Again, this is due to the model being seeded by a cluster of small motions at the start, which causes the model to make a general guess at which way the action will go.



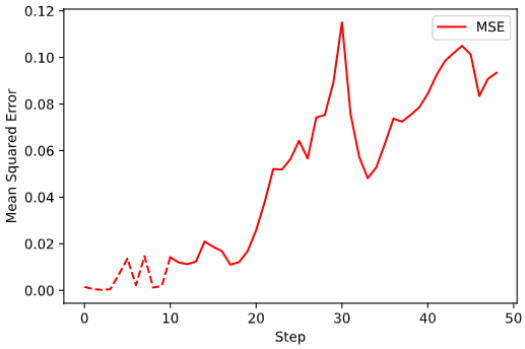
(a) High AUC and High MSE 1



(b) High AUC and High MSE 2



(c) MSE. Step: 5



(d) MSE. Step: 10

Figure 4.8. Example High AUC and High MSE Ground Truths And Predictions.

Both these trajectories are examples of motions that could be detected as being choices of low confidence. These results show that it is indeed possible to see low confidence with this

type of model. However, as the next section will discuss, the neural network can predict future trajectories accurately.

4.7 Inaccurately Measured Conflicted Trajectories

The final category to explore is where the prediction coincides with the ground truth indecisive examples, which results in a low MSE and a high AUC. This result means a higher confidence measurement, which is not representative of the trajectory. Because the model was trained on a wide variety of subjects and mouse-motions, it is natural to replicate them when given the correct stimuli. With the earlier cases, 490 trials matched the high AUC scores and low MSE scores, leading to several interesting patterns of accurate prediction. While these are edge cases, understanding them can expose model weaknesses and inform on future methodologies.

Statistics in Tables 4.9 and 4.10 reinforce the issue. Because the forward MSE stays consistently low, the model's overfit performance means little detection of choice conflict, contrary to what the AUC suggests. These forward predictions are generally shorter, as seen from the prediction length, but the trials' sizes are usually longer.

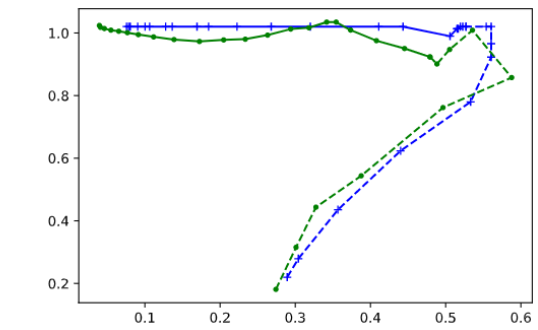
Table 4.9. High AUC and Low MSE By Test Set

	MSE	Coordinate MSE	Forward MSE	Forward Prediction Length
mean	0.001750	0.002544	0.002426	20.359184
std	0.000766	0.002182	0.000964	12.448568

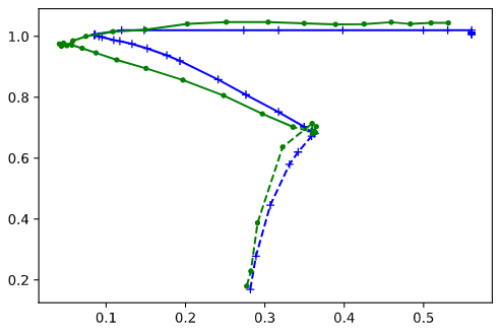
Table 4.10. Correlation Between Error and Metrics for High AUC and Low MSE Ground Truths And Predictions.

Step	-1	5	10	30	50	75	100
	MSE						
AUC	0.047653	-0.013772	-0.010123	-0.101958	0.149861	0.13486	0.222468
MD	-0.07503	-0.092901	-0.180198	-0.123433	-0.076921	0.034501	-0.409254
	Coordinate MSE						
AUC	0.169297	-0.40405	0.100097	0.015516	0.250253	0.213964	-0.036979
MD	0.067916	0.181471	-0.163342	0.041305	0.079943	0.144367	-0.064376
	Forward MSE						
AUC		-0.01624	0.047922	-0.035937	0.055223	0.374901	0.008872
MD		-0.064836	0.005762	0.068138	-0.05895	0.012712	-0.098395

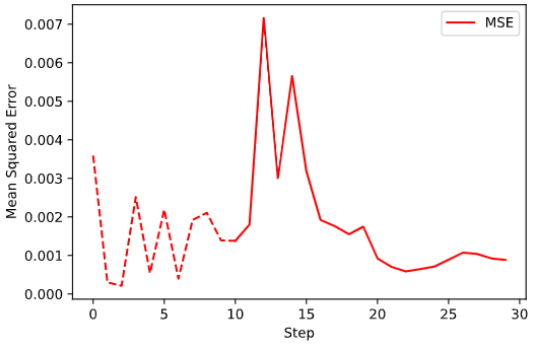
Both cases in Figure 4.9 are seeded at equal lengths, but the left prediction does not see the turn in the forecast but still sees accurately predicts that the motion will shift to the right. This suggests that a model can mimic unconfident behavior based on only a few seed steps. In Figure 4.9c, the counter is seen. The model has seen where the direction will go and therefore follows it relatively accurately until it gets to a typical ending position, which then switchbacks towards the other option. The trajectories have similar metrics in both samples, but because of differences in the seeding sequence, having more time spent before significant results in more predictive behavior.



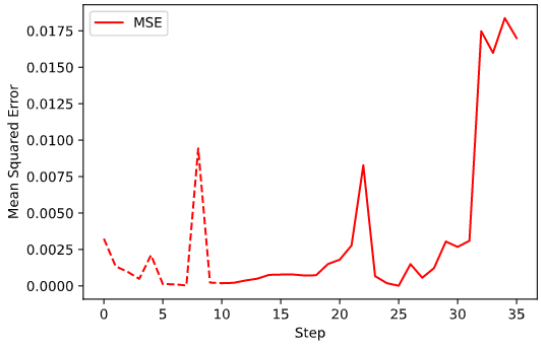
(a) High AUC and Low MSE 1



(b) High AUC and Low MSE 2



(c) MSE. Step: 10



(d) MSE. Step: 10

Figure 4.9. Example High AUC and Low MSE Ground Truths And Predictions.

4.8 Results Discussion

Every case study highlights distinctive features and their performance and shows the predictive and evaluative measures of LSTMs in this application. Short, direct trajectories, typically the most confident, will be easier to evaluate as confident. In comparison, longer trajectories will generally be more challenging to get accurate results. However, this can be interpreted as a sign of confidence, informing the model's hidden state as to what direction is more likely. This feature is especially true when more time and little motion is spent near the starting position.

The measure of future predictions shows potential in confidence measurement, with further study of this novel approach will result in a clearer understanding of its limits and

capabilities. Further exploring metrics such as speed and time deltas could also improve the accuracy of results, as here they became minimized compared to other metrics. Metrics such as position and position deltas supply exciting feedback to the model, and their importance in the prediction evaluation is evident by these results.

LSTMs are capable of delivering an accurate measure of confidence on mouse trajectories. There are a variety of circumstances in which a model's prediction can be related to other confidence metrics of mouse motion. Confident motions that do not contain extended pauses or idling movements will result in a forecast with low error, which can be interpreted similarly to original confidence metrics. However, mouse motions with extended delays are difficult to predict accurately. Similarly, when a trajectory includes complex motion that results in lower confidence, the prediction error can be higher. There are also low confidence trajectories that the model can follow with little context, likely due to overfitting. Further study of confidence estimation with neural networks will broaden the understanding and application of this technique. By utilizing the sequential predictive ability of LSTMs to evaluate a likely path of a mouse trajectory, valuable statistics can be gathered to develop individuals' fuller profiles.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion

As shown, LSTMs are capable of delivering an accurate measure of confidence in this use case. It is seen that there are a variety of circumstances in which a model's prediction can be related to other confidence metrics of mouse motion. Confident motions that do not contain extended pauses or idling movements will result in a forecast with low error, which can be interpreted similarly to AUC and MD's original metrics. However, when the trial contains extended pauses, it becomes more difficult for a trajectory to be predicted accurately. Similarly, when a trajectory includes complex motion that results in higher AUC and MD values, the prediction error can be higher. However, there are also low confidence trajectories that the model can follow with little context, likely due to overfitting. More study of confidence estimation with neural networks will broaden the understanding and application of this methodology. By utilizing the sequential predictive ability of LSTMs to evaluate a likely path of a mouse trajectory, valuable statistics can be gathered to develop individuals' fuller profiles.

Actionable intelligence can be informed using models of this nature to understand subjects better. By developing a clear understanding of neural network model capabilities, leaders can create better-informed security policies to protect individuals and organizations. There are two directions researchers can explore. A proactive one leverages neural networks like LSTMs to screen for conflictive behavior among potential applicants and personnel to identify subjects that pose security risks. The defensive direction seeks to protect trusted individuals from being targeted by external threats seeking to exploit social vulnerabilities. Both security postures can reinforce existing measures meant to identify and secure against social engineering threats. Psychological focused research, such as the ones this work builds from, can also benefit from an expanded toolset in analyzing and studying subject behavior with flexible tools that are deployable in various environments.

5.1 Future Work

This work opens the door for more targeted approaches to understanding this field of HCI. A variety of experiments by machine learning and security researchers will develop this

intriguing subject. These ideas look to expand the study of human confidence measurement to become a new avenue for machine learning. Of note, this approach's anonymous technique disconnects from the need to build user profiles, as cognitive confidence processes belong more to the subconscious and exist more uniformly in people [46]. Metrics of this nature provide a more in-depth insight into a person's mental activity. While it likely cannot detect falsehoods explicitly, it can indicate the need for further investigation. It could also be deployed to measure competency on a system, allowing administrators to picture users' mental state better. Detection of behavioral change could indicate users reacting to a sudden unexpected change to a network, acting as an early indicator of issues for administrators.

The first is, can a website be developed, which can measure user confidence? Some approached worth exploring approaches are active, where a user answers questions directly and provides feedback to score the confidence of their answer. Another is a passive one, which collects mouse motions from a user as they navigate a webpage or site with a given goal and then inquiring about their state of mind afterward. The goal is to train models on mouse data that is more characteristic of real-world examples.

Second, what are the different characteristics and features of different qualities, distinguishing between confident and unconfident, truth-tellers and liars, and those impersonating others? While there is a significant overlap of each type of behavior, a person pretending to be someone else on a network is lying about their identity and may respond to inquiries with little confidence. When they are distinct from one another, identifying these differences will make for more robust profiling. Someone unsure of their commitments is not lying in responses but still register as dishonest when, in reality, they are susceptible to cohesion.

The third question is, how can countermeasures be applied to protect online activity from being analyzed for confidence evaluation? Possible solutions are employing fuzzing techniques to obscure a person's confidence and identity effectively. Important goals would be to ensure that the methods used covert methods not to disrupt user activity. Applying this may require additional explorations of confidence metrics beyond mouse trajectory analysis, like keyboard dynamics.

What happens when predictions are made from the beginning and ending positions of a sequence? The model measures the most likely path between the two points and then measures them against the ground truth. This approach mimics calculating AUC and MD,

except it can be used continuously on active mouse motion, breaking up a sequence in various ways, such as by time or between anchor points along cursor's movement, such as buttons on a webpage. Approaching inference on discrete points may be the solution to deploying this approach in live scenarios.

With consideration of keyboard dynamics, another question arises. Can confidence be inferred from the use of a keyboard? In some cases, the mouse is not used to express a decision but instead written out, which means exploring typing information, similar to lie and authorization detection.

Can internal and external threats be identified using confidence analysis? One possible approach would be to prompt questions to select benign and hostile agents and develop a classifier that can use mouse trajectories to determine which might be malicious by detecting lies and confidence. A hostile actor may answer honestly, but hesitation is more expressive in confidence measurement and not lie detection because of their intentions.

Another security application to explore is having measurements done on individuals as they are introduced and acclimate to a new digital environment. The goal would be to measure how their confidence changes as they become accustomed to the environment's design and if it changes throughout their use. This experiment can also be approached similarly to the previous question, detecting new users to a system, anonymously by device, or tracking specific users. Detection of new users could alert administration of a new, unaccustomed user, and may want to employ educational techniques to improve familiarity with the new system or investigate if access from a new user is not expected.

Finding answers to these questions would be valuable to employ various machine learning architectures to determine which behaves best. Beyond LSTMs, there are GRUs, which behave similarly but are more straightforward in design and have comparable performance, making deployment less resource-intensive. CNNs are also potential candidates as they are well suited to identifying patterns using sliding window techniques. Other designs of an LSTM model may also improve results and performance, such as bidirectional LSTMs.

5.2 Final Thoughts

The evaluation of confidence still holds many questions, but this work serves as a groundwork for developing more robust solutions. It finds there are recognizable patterns to confident and indecisive mouse trajectories answered honestly. Exploring the given case studies of mouse data against the predictions of a trained LSTM neural network shows several features that are characteristic of confident and unconfident behavior. Namely, smooth direct motion with no idling movements predominately features confident trajectories, which a model can determine as such due to the short span of activity and directness. However, once courses take more time, it becomes more challenging to measure confidence accurately against the ground truth measurements. In some cases, overfitting caused predictions that were similar to ground truth measurements, making unsure motions appear to be confident.

Our growing interaction with computers and ever-increasing connection to the Internet exposes how we use them potentially to everyone. It is essential to know what is learnable from these seemingly benign statistics. Others could exploit them to achieve their goals—in reverse, using this data to defend against antagonistic forces exploiting any weaknesses available to them. Lastly, understanding what private entities can do with this information will better inform the necessity of robust privacy measures online to protect the general public.

List of References

[1] S. C. Woolley and P. N. Howard, *Computational Propaganda: Political Parties, Politicians, and Political Manipulation on Social Media*. Oxford, UK: Oxford University Press, 2018.

[2] R. Faris, H. Roberts, B. Etling, N. Bourassa, E. Zuckerman, and Y. Benkler, "Partisanship, propaganda, and disinformation: Online media and the 2016 US presidential election," *Berkman Klein Center Research Publication*, vol. 6, 2017.

[3] M. Monaro, L. Gamberini, and G. Sartori, "The detection of faked identity using unexpected questions and mouse dynamics," *PLOS One*, vol. 12, no. 5, p. e0177851, 2017.

[4] K. Revett, *Behavioral Biometrics: A Remote Access Approach*. Hoboken, NJ, USA: John Wiley & Sons, 2008.

[5] P. Zimmermann, S. Guttormsen, B. Danuser, and P. Gomez, "Affective computing—a rationale for measuring mood with mouse and keyboard," *International Journal of Occupational Safety and Ergonomics*, vol. 9, no. 4, pp. 539–551, 2003, pMID: 14675525. Available: https://doi.org/10.1080/10803548.2003.11076589

[6] T. Yamauchi, "Mouse trajectories and state anxiety: Feature selection with random forest," in *Proc. 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, Sep. 2013, pp. 399–404.

[7] A. Kaklauskas, "Web-based biometric computer mouse advisory system to analyze a user's emotions and work productivity," *Intelligent Systems Reference Library*, vol. 81, pp. 137–173, Dec. 2015.

[8] D. Sun, P. Paredes, and J. Canny, "MouStress: Detecting stress from mouse motion," in *Proc. Conference on Human Factors in Computing Systems*, Apr. 2014.

[9] A. Ribeiro, D. Carneiro, P. Novais, and J. Neves, "Monitoring mental fatigue through the analysis of keyboard and mouse interaction patterns," in *Proc. International Conference on Hybrid Artificial Intelligence Systems*, Sep. 2013, pp. 222–231.

[10] E. Hehman, R. M. Stolier, and J. B. Freeman, "Advanced mouse-tracking analytic techniques for enhancing psychological science," *Group Processes & Intergroup Relations*, vol. 18, no. 3, pp. 384–401, 2015.

[11] J. B. Freeman and N. Ambady, "MouseTracker: Software for studying real-time mental processing using a computer mouse-tracking method," *Behavior Research Methods*, vol. 42, no. 1, pp. 226–241, 2010.

[12] P. E. Stillman, X. Shen, and M. J. Ferguson, "How mouse-tracking can advance social cognitive theory," *Trends in Cognitive Sciences*, vol. 22, no. 6, pp. 531–543, 2018.

[13] S. Garcia-Guerrero, D. O'Hora, A. Zgonnikov, and S. Scherbaum, "The action dynamics of approach-avoidance conflict in decision-making: A mouse-tracking study," May 2019, unpublished. Available: psyarxiv.com/4658p

[14] R. Shikder, S. Rahaman, F. Afroze, and A. A. Al Islam, "Keystroke/mouse usage based emotion detection and user identification," in *Proc. 2017 International Conference on Networking, Systems and Security (NSysS)*. IEEE, 2017, pp. 96–104.

[15] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proc. of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security* (VizSEC/DMSEC '04). New York, NY, USA: Association for Computing Machinery, 2004, p. 1–8. Available: https://doi.org/10.1145/1029208.1029210

[16] N. Abd Hamid, S. Safei, S. D. M. Satar, S. Chuprat, and R. Ahmad, "Mouse movement behavioral biometric systems," in *Proc. 2011 International Conference on User Science and Engineering (i-USEr)*. IEEE, 2011, pp. 206–211.

[17] Z. Jorgensen and T. Yu, "On mouse dynamics as a behavioral biometric for authentication," in *Proc. of the 6th International Symposium on Information, Computer and Communications Security, ASIACCS 2011*, Jan. 2011, pp. 476–482.

[18] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, and A. Schclar, "User identity verification via mouse dynamics," *Information Sciences*, vol. 201, pp. 19 – 36, 2012. Available: http://www.sciencedirect.com/science/article/pii/S0020025512001946

[19] C. Shen, Z. Cai, X. Guan, Y. Du, and R. Maxion, "User authentication through mouse dynamics," *Information Forensics and Security, IEEE Transactions on*, vol. 8, pp. 16–30, Jan. 2013.

[20] B. Sayed, I. Traoré, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013.

[21] S. Mondal and P. Bours, "Continuous authentication using mouse dynamics," in *Proc. 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*. IEEE, 2013, pp. 1–12.

[22] M. Sauerland and S. Sporer, "Post-decision confidence, decision time, and self-reported decision processes as postdictors of identification accuracy," *Psychology*, vol. 13, no. 6, pp. 611–625, Dec. 2007.

[23] R. Moreno-Bote, "Decision confidence and uncertainty in diffusion models with partially correlated neuronal integrators," *Neural computation*, vol. 22, pp. 1786–811, Feb. 2010.

[24] A. Insabato, M. Pannunzi, E. T. Rolls, and G. Deco, "Confidence-related decision making," *Journal of Neurophysiology*, vol. 104, no. 1, pp. 539–547, 2010.

[25] M. N. Hebart, Y. Schriever, T. H. Donner, and J.-D. Haynes, "The Relationship between Perceptual Decision Variables and Confidence in the Human Brain," *Cerebral Cortex*, vol. 26, no. 1, pp. 118–130, Aug. 2014. Available: https://doi.org/10.1093/cercor/bhu181

[26] W. De Neys, S. Cromheeke, and M. Osman, "Biased but in doubt: Conflict and decision confidence," *PLOS One*, vol. 6, no. 1, pp. 1–10, 2011.

[27] J. Freeman, R. Dale, and T. Farmer, "Hand in motion reveals mind in motion," *Frontiers in Psychology*, vol. 2, p. 59, Apr. 2011.

[28] J. Navajas, M. Sigman, and J. E. Kamienkowski, "Dynamics of visibility, confidence, and choice during eye movements," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 40, no. 3, p. 1213, 2014.

[29] A. Lak, G. Costa, E. Romberg, A. Koulakov, Z. Mainen, and A. Kepecs, "Orbitofrontal cortex is required for optimal waiting based on decision confidence," *Neuron*, vol. 84, no. 1, pp. 190–201, Oct. 2014.

[30] A. Boldt and N. Yeung, "Shared neural markers of decision confidence and error detection," *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, vol. 35, pp. 3478–84, Feb. 2015.

[31] M. Monaro, F. I. Fugazza, L. Gamberini, and G. Sartori, "How human-mouse interaction can accurately detect faked responses about identity," in *Symbiotic Interaction*, L. Gamberini, A. Spagnolli, G. Jacucci, B. Blankertz, and J. Freeman, Eds. Cham, Switzerland: Springer International Publishing, 2017, pp. 115–124.

[32] M. Monaro, C. Galante, R. Spolaor, Q. Q. Li, L. Gamberini, M. Conti, and G. Sartori, "Covert lie detection using keyboard dynamics," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[33] M. Hibbeln, J. Jenkins, C. Schneider, J. Valacich, and M. Weinmann, "Investigating the effect of insurance fraud on mouse usage in human-computer interactions," in *Proc. Thirty Fifth International Conference on Information Systems, Auckland 2014*, Jan. 2014.

[34] J. S. Valacich, J. L. Jenkins, J. F. Nunamaker Jr, S. Hariri, and J. Howie, "Identifying insider threats through monitoring mouse movements in concealed information tests," in *Proc. Hawaii International Conference on System Sciences. Deception Detection Symposium*, 2013.

[35] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[36] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1025–1032.

[37] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.

[38] A. Géron, *Hands-on Machine Learning With Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[39] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019, pMID: 31113301. Available: https://doi.org/10.1162/neco_a_01199

[40] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6645–6649.

[41] C. Calluso, A. Tosoni, G. Pezzulo, S. Spadone, and G. Committeri, "Interindividual variability in functional connectivity as long-term correlate of temporal discounting," *PLOS One*, vol. 10, no. 3, pp. 1–22, 2015.

[42] C. Calluso, L. Cannito, A. Tosoni, M. L. Carenti, G. Martinotti, and G. Committeri, "Intertemporal choice behavior and decision dynamics in pathological gamblers," presented at Ten years of Mind/Brain Sciences at the University of Trento, Oct. 2017.

[43] C. Calluso, A. Tosoni, L. Cannito, and G. Committeri, "Concreteness and emotional valence of episodic future thinking (EFT) independently affect the dynamics of intertemporal decisions," *PLOS One*, vol. 14, pp. 1–22, May 2019.

[44] C. Calluso, M. A. Zandi, and M. G. Devetag, "Cognitive dynamics of religiosity and intertemporal choice behavior," *Journal of Cross-Cultural Psychology*, vol. 51, no. 9, pp. 719–739, 2020.

[45] N. F. Lepora and G. Pezzulo, "Embodied choice: How action influences perceptual decision making," *PLOS Computational Biology*, vol. 11, no. 4, pp. 1–22, Apr. 2015. Available: https://doi.org/10.1371/journal.pcbi.1004110

[46] C. Calluso, G. Committeri, G. Pezzulo, N. Lepora, and A. Tosoni, "Analysis of hand kinematics reveals inter-individual differences in intertemporal decision dynamics," *Experimental Brain Research*, vol. 233, no. 12, pp. 3597–3611, 2015.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California